A method to record and predict human imagination based on advansed bidirectional heuristic

Louie(Yi) Lu
Louie.lunz@gmail.com

***ABSTRACT*:**
This paper provides a new perspective which is making human associational path quantizable and recordable, in other words, it is a method to learn human self-association by the cognition of features in vocabulary.

KEYWORDS:
Bidirectional Heuristic, generalized *Mahalanobis* distance, *Jaccard* similarity coefficient, natural language processing, associational path.

1. Introduction

In order to understand human consciousness, the researchers from around different scientific fields are seeking a measuring method of human consciousness. Hence, there are various ways to measure human consciousness such as neural correlates, behavioral correlates and creative correlates, etc. but there is no final conclusion about human consciousness among psychology, brain's neurology.

Following [2], in the Chapter 4, Jeff mentioned that the neocortex has three major features **(sequential storage, self-associational memory and constant characteristic) t**hat comprise as essential elements for predicting the feature based on the memory from the past. According to the above, I propose a method that can let the computer measure and quantify the predicted ability of human consciousness.

Therefore, in this paper, our method is to find out the similarity and regularity through human associative ability, and then to develop a network of association which consist of certain associational paths. The network of associations above, it's a perspective to help both human and computer to understand human consciousness.

CASE 1.1. For instance, if a computer asks a question, what comes to your mind when you react to this following statement: Thick layers of dark clouds. And someone's answer is an abstract noun "hope". In fact, the dark cloud tends to imply negative emotion, and "hope" seems to be more positive. So, even a human does not necessarily likely to guess about this logical relationship between "cloud" and "hope", but some people may understand or infer that his self-association is likely to achieve "silver lining".

Hence, If the computer could find out the key word “light” which is connected between “cloud” and “hope”, then the computer would find out a rational **associational path** about “silver lining”**.** This paper proposes a method about how to extract the common feature among these words, and generate the path.

The **Section 2** introduce the overall of methodology which including the architecture and process. The main algorithms implementation will show in **Section 3**..

2. Methodology

In this section, I introduce the architecture of the method about how to quantify human associational path and the reason why I’ve chosen the related algorithms. The creation and idea of this method is corresponding to the following questions.

Following [2] mentioned, in order to make sequential storage, self-associational memory and constant characteristic to be quantizable, we have three following preconditions:

1. All the memory which stored in brain can express through constant characteristic words.
2. Each word could be described by several quantizable adjective and character.
3. Association is based on the memory both present and past.

Because of human behaves are not absolutely reasonable, and the decision made by human largely depends on their constant characteristic information and logical relationship that stored in their brain. Therein lies the problem, it is possible to generate the wrong decision for lack of information and some logic rules, or misunderstanding of information and faulty logic which are affected by emotion and irrelevant self-association.

For solving above problem, we set the data input as objective constant characteristic information (vocabulary database) and subjective information (words from individual person’s description), and the goal is to measure emotion and self-association, finally to map individual people’ subjective logic, collecting all of these logics to generate a network.

Following **CASE 1.1,** the word “cloud” is a countable concrete noun, and “hope” is an abstract noun. Because we cannot unify each feature from the elements in X, so It’s hard to compare their relationship. Hence, we need to extract the feature from the word “hope”, and compare with the feature from “cloud”.

there are four reasons why we have to choose **Bidirectional Heuristic Search** as following.

1. Because the goal node might have no similar feature with the start node, so we need to search number of the closest neighbors around goal node for finding out the similar feature between start and goal node.
2. We need to form a path, the direction between start and goal node need to head for a middle node with the most significant common feature. Therefore, it has to start from both sides' node.
3. We choose front-to-front which is making the next node from start or goal node to be the new goal node for next search. it will increase the efficiency and probability of the first collision.

2.1 Overall architecture

As shown on **Fig 2.1**, There are four major parts in our system.

1.The blue solid box is the database; the set Y is defined as quantitative abstract emotion. The set X is defined as constant characteristic nouns.

2.The blue solid rounded circle is metric distance learning, greed ball means the samples in X have the same label Y, red ball means their samples have the similar labels, the blue ball means have no similar labels.

3.The elliptical dashed circle is associational path by bidirectional heuristic search.

4.The blue dashed circle is multi-paths overlapped.

Because the initial database of Part 1 is self-defined, it complied with objective logic, but not precise. So, the learning process is to run from **Part 1** to **Part 4** iteratively in order, and then corrected parameter for **Part 1** and get an associational logic link in **Part 4** eventually. we want to learn human's subjective logic of their own self-association, so we cannot preset or default any logic, and there is no direct correspond relationship for each word as well. Our method is for separating the objective causal relationship, subjective cognition, and subjective attitude and emotion.

The input data of this method is from English dictionary and latest semantic definition by different fields of science. We should quantify the corresponding adjective to describe the feature of the noun in set X, also the quantized emotion value in the set Y, and the quantized value is self-defined. Hence, in order to let the self-defined quantized value be more accurate, the data which is collecting from people should be in some specific scenario. After we use lots of cases to interact with human, following the Part 4 is keep coming back to part 1 reclusively and iteratively, in the end, the output is a trained database from part 1 and a logical path with weighted from Part 4.

Following **CASE 1.1,** some people might answer "It's going to rain" or "doomsday", the meaning of part 4 is to record the answerers thinking model and the degree of self-association. For a same person, the computer will predict his answer in similar case next time.

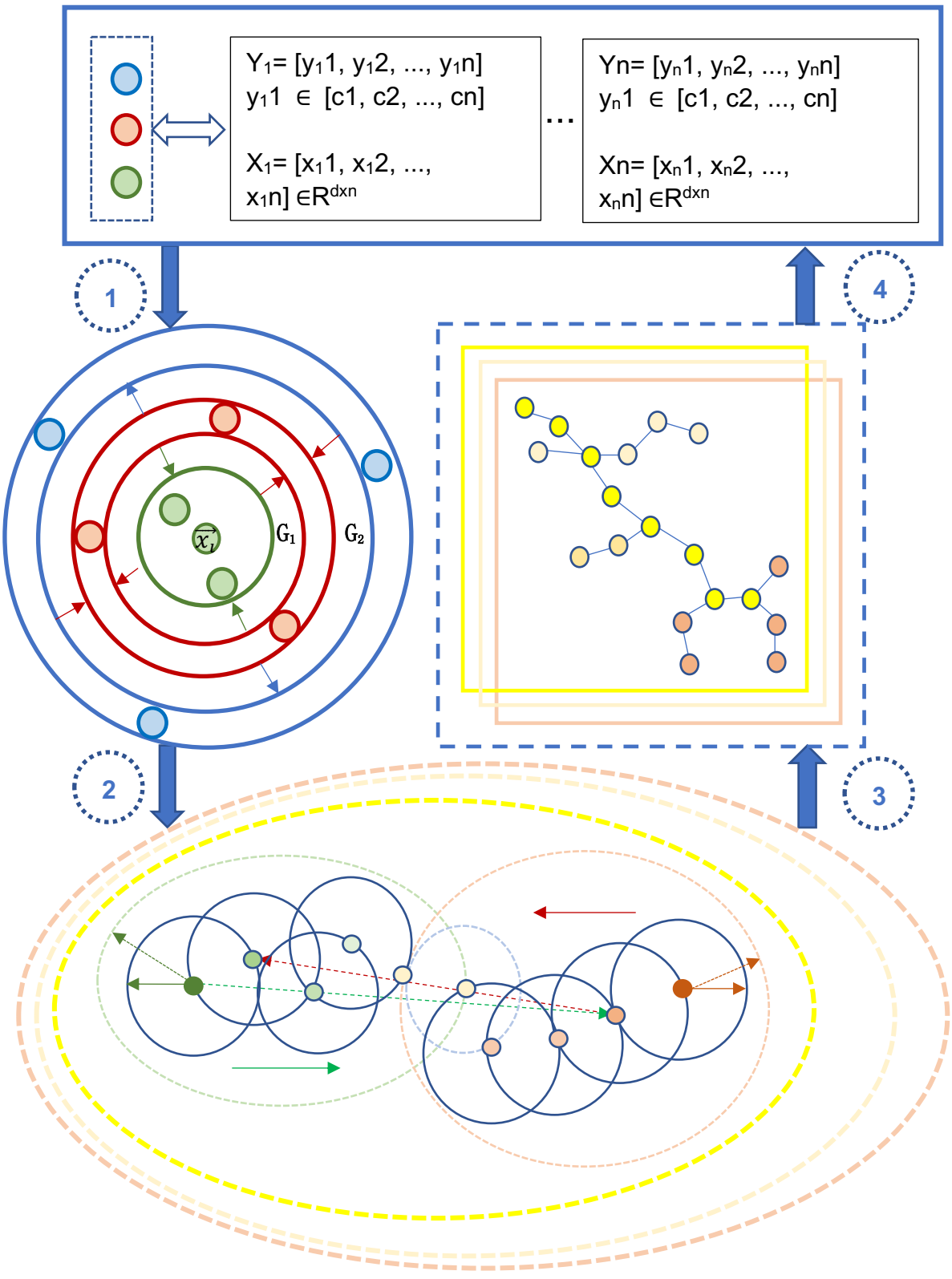
Y_1= [$y_1$1, $y_1$2, ..., y_1n]
$y_1$1 ∈ [c1, c2, ..., cn]
X_1= [$x_1$1, $x_1$2, ..., x_1n] ∈R^{dxn}
...
Yn= [y_n1, y_n2, ..., y_nn]
y_n1 ∈ [c1, c2, ..., cn]
Xn= [x_n1, x_n2, ..., x_nn] ∈R^{dxn}
1
2
3
4
$\vec{x_i}$
G_1
G_2

Fig.2.1 Method Outline

2.2 Process

The process of the method has five steps as following.

Step 1: Database X and Y construction

This is the most significant part, and need to be recursive and iterative. **"Y_i"** are the label set of **"Xi"** which correspond to different kinds of attitude or emotion, such as the level of positive, happy, angry, ...etc. Following **Case 1.1,** the known condition is a statement "Thick layers of dark clouds", and the subjective associational answer is "hope". Hence, the data processing is need to comply the following constraint. As shown on **Table 2.1**,

In this case, we only need a set of **Xi**, and use the level of positive to be the label **Yi**. We need to define comprehensive and quantitative features for each noun, make the key word "cloud" as **"x_i"** in the set **"Xi"**, **"f_i"** is the adjective feature of **"xi"** such as thin to thick, number of layers and color, etc. **"fq_i"** is quantitative value of **"fi",** and **"w_i"** is the weight of **"f_i"**. and **"fq_i"** is quantitative value of **"fi"**. we also set the variable labels of emotion as set **Yi**, in this case, **"y_i"** is the supervise label of **"xi"** and defined as positive and negative emotion, **"ci"** is the adjective of emotion such as struggling, fighting, etc. **"cq_i"** is the quantitative value of **"ci"**.

X	X_1	$x_1$1	$f_1$1	$w_1$1 * $fq_1$1
			...	
			f_1n	w_1n * fq_1n
		...		
		x_1n	$f_1$1	$w_1$1 * $fq_1$1
			...	
			f_1n	w_1n * fq_1n
	Xi	**x_i**	**f_i**	**w_i** **fq_i**
	Xn	x_n1	f_n1	w_n1 * fq_n1
			...	
			F_nn	w_nn * fq_nn
		x_i, x_j, ...		
		x_nn	f_n1	w_n1 * fq_n1
			...	
			f_nn	w_nn * fq_nn

Y	Y_1	$y_1$1	$c_1$1	$cq_1$1
			...	
			c_1n	cq_1n
		...		
		y_1n	$c_1$1	$cq_1$1
			...	
			c_1n	cq_1n
	Yi	**y_i**	**c_i**	**cq_i**
	Yn	y_n1	c_n1	cq_n1
			...	
			c_nn	cq_nn
		y_i, y_j, ...		
		y_nn	c_n1	cq_n1
			...	
			c_nn	cq_nn

Table.2.1 databased table

- **Initialized the database**

Following **Table 2.1,** in order to establish the quantizable feature for each word, we need to define the a certain degree to the feature. But what would be unique about these quantitative definitions, is that we follow the degree of the average person's vague cognition to the feature. For instance, let a person to describe the color of a real apple, by contrast a whole chromatography, the describable words from a common person would be in a very small scale. Hence, the color feature of an apple could be defined in a small range 1 to 10 for **data quantification**, and each number stand for a specific color. What if there is a spot on the apple, this spot is a feature as well, but the weight is relatively small compared to the whole feature, so we use "**wi**" to weight the spot feature, and the whole feature of an apple is the weighted average. As a result, each degree of a feature is expressible, explicable, and describable by average person, because our goal is for measuring human subjective cognition. The definition of an emotion is the same, **"ci"** is an adjective of an emotion, there may be several adjective for describing one emotion. A sample word **"xi"** may contain several emotions **"yi"**.

- **Original data collection and decomposition**

The original data set **X** are like a huge information ocean, Following **CASE 1.1,** even though we know the two key words are "cloud" and "hope", we cannot measure all the distance between a word and a whole dictionary. Therefore, we need only to collect the relevant data. Hence, the first step is to input all the **"xi"** with **negative and positive label "yi" to be the set Xi and set Yi.**

- **Data normalization**

The quantified features' number should be reduced dimensions via principal component analysis (PCA), and normalized from 0 (inclusive) to 1 (inclusive)

Step 2: Algorithm 3.1: Multi-labels Supervised Metric Learning (MSML)

In the **CASE 1.1**, we set all the words as set **X**, and **$X= [x_1, x_2, ...x_n]$**. The word "cloud" and "hope" as x_i and x_j, the constant characteristics of x_i are its features which could be quantified such as components, color, thickness, size, shape, etc. And the above characteristics could be quantified or normalized as linear number. **$x_i= [f_1, f_2, ...f_n]$**, **f_i** is a quantized feature such as the color break down into **f1**, **f2**, and **f3**, **f1** is defined to be brightness and the range **fq1** is from 0 to 255, **f2** is tone, **f3** is saturation analogously. The "bright" is an adjective to word "future", so set it as a feature "fi". Moreover, we set the variable labels of emotion as set **Y**, and **$Y= [y_1, y_2, ...y_n]$** which is corresponding to set X. And the label set **Y** is defined to positive and negative emotion value which could measure the level of emotion, the set C is quantified adjective, **c1** is defined to a specific positive and negative emotion, cq1 is the value.

Therefore, we can see the word “cloud” convert into a symbolic vector which consist of vector x_i and y_i. For example, x_1, x_5, and x_{13} could be all same as “cloud”, but with different quantitative properties such as dark clouds, thin cloud, and thick white clouds. The yi could be a specific positive and negative value.

In order to measure the semantic distance between x_i with supervised label y_i to the whole set X and Y, we use a special **supervised metric leaning** method, the closest method is *Mahalanobis* distance, but above ready-made algorithm is not suit for our method, because yi is a set as well. Inspired by Locally Supervised Metric Learning (LSML) which is proposed by Jimeng sun at al [11], we propose a **multi-labels’ generalized *Mahalanobis* distance** learning method to handle multi-labels’ case.

Step 3: KNN Classifier for extracting the same feature in goal node

Following **CASE 1.1**, Because there is no common feature between “hope” and “cloud”, so we use KNN to get the k number of nearest neighbors around “hope”, then compare these neighbors feature with “hope”. For example, if the nearest neighbor is word “future”, there is a feature which defaulted as “bright”, then the word “future” is the first associative-word, feature “bright” is the common feature between “cloud” and “hope”. As shown on **Fig 2.2** and **Table 2.2**.

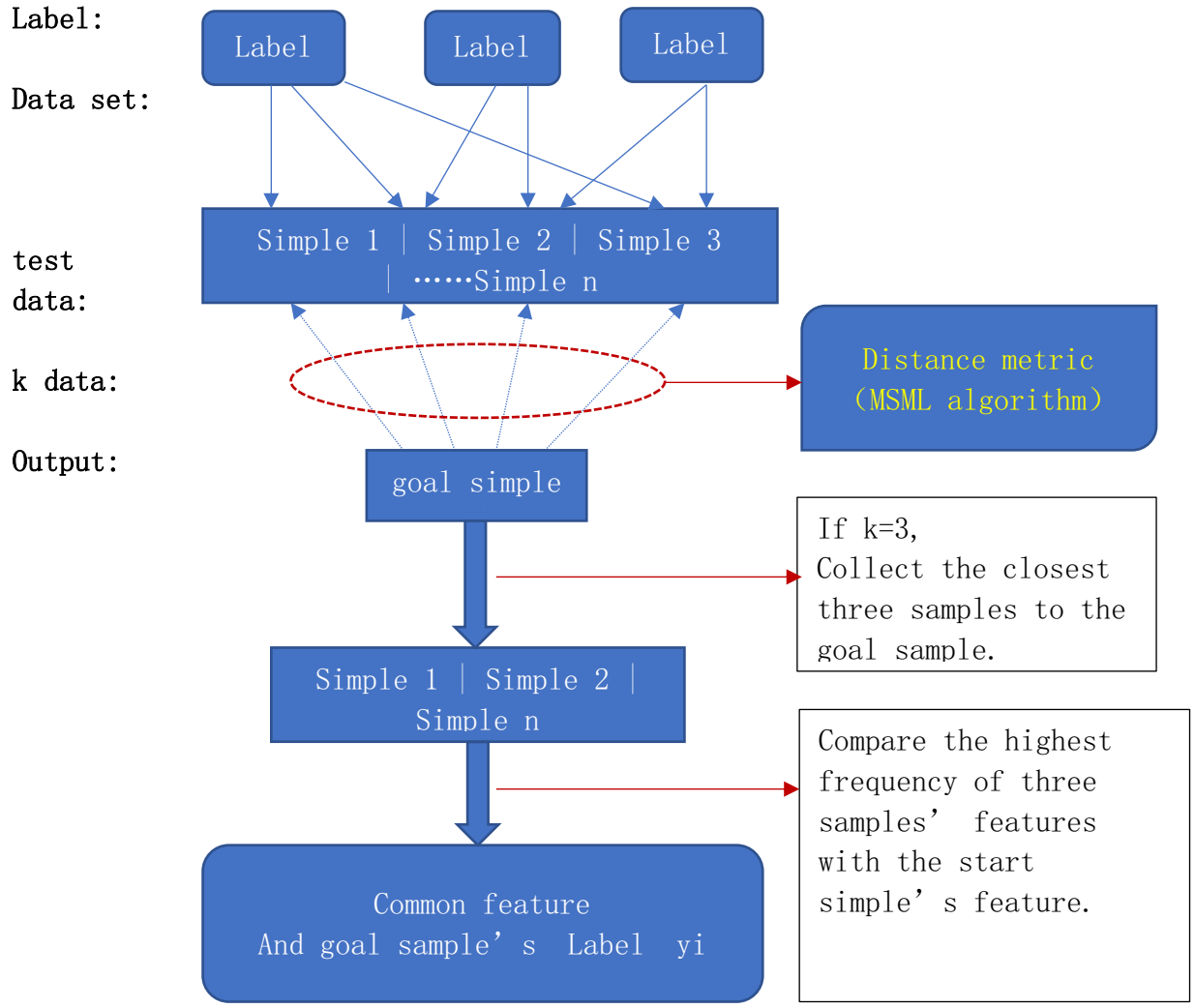

Fig.2.2 KNN Classifier

Input	Process	Output
1. Data set **X** **2.** Labels **Y** **3. Distance Metric** **4**. **k** value **5**.Test data **xj**	**1.** Calculate the distance between **xj** and each data from **X** data se**t.** **2**. Ascending numerical sort. **3**. Pick the nearest **k** neighbors. **4**. Compare the highest frequency of three samples' label, to be xi's label. **5**. Compare the highest frequency of three samples' features with the start simple's feature, get the common feature.	**1. yi** is the label for test data **xi.** **2.** common feature **fi.**

Table.2.2 KNN algorithm

Step 4: Algorithm 3.2: BFIDWS-MSML

Among the problems we encountered, there is no barrier in this data map, and basically there is no specific nodes expansive tree, because of the new node is an obscure self-association from a person, and this algorithm try to conjecture it. Therefore, we utilize the similarity of the characteristic of some certain word to create the expansion. It also means there is no absolute optimal cost, the relative optimal path should meet following conditions.

- Each new node is the nearest approach the feature to the previous node and the nearest the goal node.
- The opposite new node to be the new goal node, and the start node collect number of near nodes with the common feature from new goal node.
- All the expansive nodes could form a train of explicable and logical transformation from start node to goal node.
- Find out the minimum expansive nodes to achieve the explanation of the associative process.

Step 5: Associational path: Multi-layers roadmap overlapped

We need to build multi-layers map, and then stagger and overlap them for making a new tree map with extensional nodes, in the end pruning and generalizing the weighting for the branches. The reasons for above as following mentions.

1.For training the data set **X** and label set **Y**. After we overlapped the paths with different Label set **Y**, or different common feature extraction tactics,

it would help to optimize the feature selection and quantization, and train the data model to be more logical and rational.

2. For further research. record the rational associational paths, means to record a type of thinking model, it can accumulate for solving complex problem. As shown on **Fig 2.3, the yellow layer** is using **BFIDWS-MSML,** and this method is to compare the common feature between both side's previous node, and choose the shortest path between both sides, and the g(n) is the minimum. Obviously, this method is not enough to form a complete associational path, therefore, we use other two layers, the **wheat layer** is based on **BFIDWS-MSML** but the feature searching is different, the next node **n** or **m** picks the previous node's common feature and combine with their present node's feature to generate new word, the new word is a reference node, the opposite node use the reference node's feature, to search the k neighbors. the **jacinth layer** is based on **BFIDWS-MSML** as well**,** the feature searching is to combine the features of both-sides present nodes, for generate new word, the new words is for observation.

As shown on **Fig 2.3,** the meaning of **W1, W3, W3** is for finding the degree of correlation for the common feature. For example, In **CASE 1.1** the goal node "hope" search around its neighbors and get the word "future", because there is a common feature "bright" between the word "future" and "cloud", if "**fi"** is the feature "bright", and as shown on **Table 2.1, fi=wi*fqi,** increase **wi** to all the subsets which contains **"fi"** in the whole **set X.** as shown on **Talble 2.3,** the subset **xi** which including common feature would get closer. In **CASE 1.1**, when we can increase each common feature's **wi,** and compare with other common feature with some value of weight, if the "bright" feature's weight makes the fewer nodes expansion than other common features, it means "cloud" and "hope" is connect by relevant concept of "bright".

Label set **Y** is positive and negative emotional level				
y1=1	y2=2	y3 =3	y4 =4	y5 =5
Data set **X** is corresponding sample.				
x1= {f1, fi}	x2= {f2, f2}	x3= {fi}	x4= {f4}	x5= {fi}
fi is the element of subset **xi, fi=wi*fqi,** when we increase the **wi**:				
y2=2	y4=4	y1=1(wi)	y3=3(wi)	y5=5(wi)

Table 2.3. example of **wi** adjustment

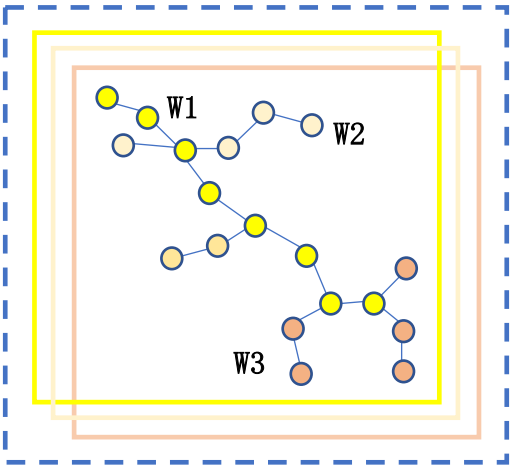

Fig.2.3 Multi-layers associational path

3. Algorithms

3.1 Multi-label supervised metric learning

To find out the sample similarity, my work is inspired by Locally Supervised Metric Learning (LSML) which is proposed by Jimeng sun at al [11], their method is based on Large Margin Nearest Neighbor (LMNN) algorithm [10] and single label supervised metric learning. But in our case, a sample will have more than one label, such as different degrees of emotion. Therefore, my algorithm improved LSML to multi-label supervised metric learning (MSML), which is considered the similarity of different samples' label

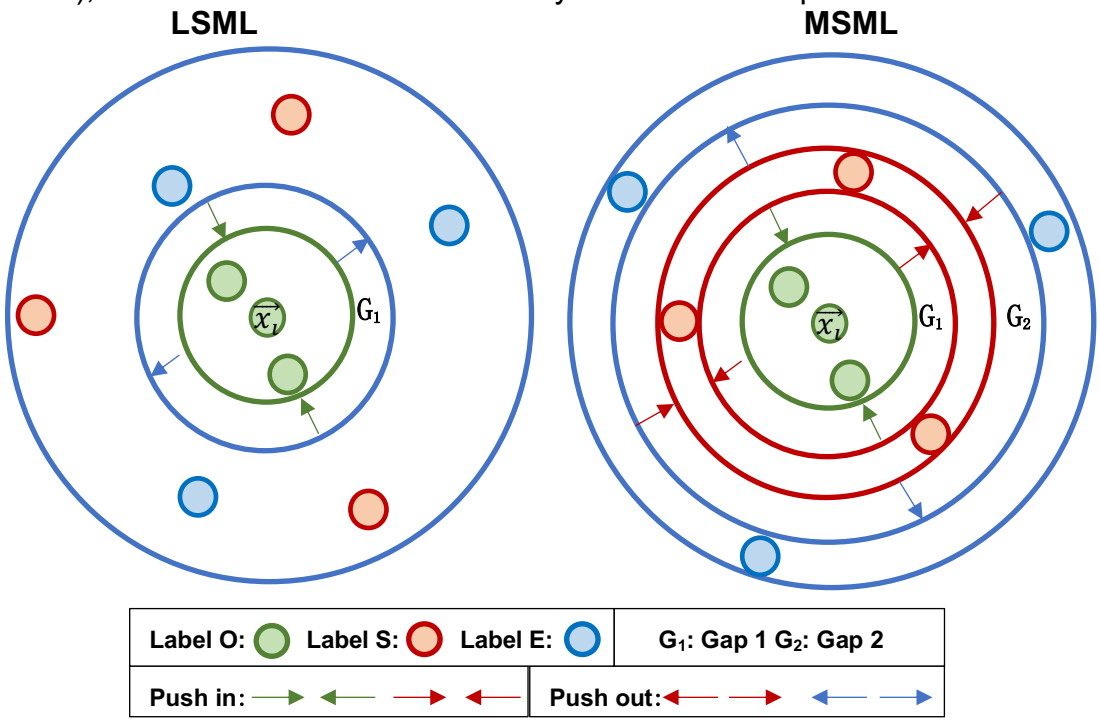

Fig.3.1 LSML and MSML distance metric

As shown as Fig.3.1, the distribution similarity of the set X would be simply divided into three groups. Label O is for some samples with same labels, Label S is samples with similar labels, Lebel Eis samples with totally different labels. MSML will constraint and tighten up the sample with similar label group, and make the gap as big as possible. Therefore, although the sample is highly similar to common labels, MSML has magnified the discrepancy and improved accuracy.

Generalized *Mahalanobis* distance

We present a multi-label supervised metric learning algorithm (MSML). Using $X = [x_1, x_2, \dots, x_n] \in R^{d\,x\,n}$ as a feature matrix of a set of samples, and $Y = [y_1, y_2, \dots, y_n]^T \in R^n$, and $y_i \in [C_1, C_2, \dots, C_n]$ as the label of x_i,
The input of KNN need X, Y and distance metric, therefore,
We need to learn a generalized *Mahalanobis* distance which like LSML[12].
Set $M \in R^{d\,x\,d}$ as a Symmetric Positive Semi-Definite (SPSP)

$$D_M(\overrightarrow{x_i}, \overrightarrow{x_j}) = \sqrt{(\overrightarrow{x_i} - \overrightarrow{x_j})^T M (\overrightarrow{x_i} - \overrightarrow{x_j})} \qquad (1)$$

In standard *Mahalanobis* distance, M is covariance matrix. But it cannot meet this particular required precision, so we need to reconstruct M.

Following [11,12], we need to define the Homogeneous Neighborhood and Heterogeneous Neighborhood, but our difference and advanced feature is that we also define a Similar Neighborhood.

DEFINITION 2.1. The homogeneous neighborhood of x_i, denoted as $\boldsymbol{N_i^o}$, is the $|N_i^o|$-nearest data points of x_i with the same labels.

DEFINITION 2.2. The similar neighborhood of x_i, denoted as $\boldsymbol{N_i^s}$, is the $|N_i^S|$-nearest data points of x_i with the similar labels.

DEFINITION 2.3 The heterogeneous neighborhood of x_i, denoted as $\boldsymbol{N_i^E}$, is the $|N_i^E|$-nearest data points of x_i with the different labels.

As shown on DEFINITION 2.2, what is similar label?

If there are two samples, the labels of sample A as y_a = [c1, c2, c3], and the labels of sample B as y_b = [c2, c3], we use Jaccard similarity coefficient to remap the label of y_a and y_b, then assign the new value to them, if their value are very close, they are similar labels.

$$\mathrm{J}(y_a, y_b) = \frac{|y_a \cap y_b|}{|y_a \cup y_b|} \quad (0 \leq J(y_a, y_b) \leq 1) \qquad (2)$$

When J =0, A and B is heterogeneous neighborhood; when J=1, they are homogeneous neighborhood. In this case, J ≥ 0.67, they are similar neighborhood. After that, each sample will have a new label value.

Following [12], we use | • | to denote set cardinality. define the local compactness measures around a feature vector $\overrightarrow{x_i}$ as:

$$O_i = \sum_{j:\overrightarrow{xj} \in N_i^o} D_M(\overrightarrow{x_i}, \overrightarrow{x_j})^2 \qquad (3)$$

$$S_i = \sum_{k:\overrightarrow{xk} \in N_i^s} D_M(\overrightarrow{x_i}, \overrightarrow{x_k})^2 \qquad (4)$$

$$E_i = \sum_{l:\overrightarrow{xl}\in N_i^i} D_M(\overrightarrow{x_\iota}, \overrightarrow{x_l})^2 \qquad (5)$$

In this case, our goal is to make the gap bilateral the similar label as big as possible, so we the gap of two sides as:

$$G_1 = \sum_{i=1}^{n}(S_i - O_i) \qquad (6)$$

$$G_2 = \sum_{i=1}^{n}(E_i - S_i) \qquad (7)$$

To let the similar sample be more compact, and irrespective sample be farther, we set the following criterion, $\boldsymbol{\sigma}$ $(0 \le \boldsymbol{\sigma} \le 1)$ is weighting parameter which is for adjust the error.

$$J = -\boldsymbol{\sigma}G_1 - (1-\boldsymbol{\sigma})G_2$$

$$= \sum_{i=1}^{n} \boldsymbol{\sigma}(O_i - S_i) + (1-\boldsymbol{\sigma})(S_i - E_i) \quad (8)$$

As M is SPSD, we factorize it with incomplete *Cholesky* decomposition, make M to triangular matrix W:

$$M = WW^T \qquad (9)$$

According to matrix trace property, and rewrite Eq.(2), (3), (4) as:

$$O_i = \sum_{j:\overrightarrow{x_J}\in N_i^O} D_M\ (\overrightarrow{x_\iota}, \overrightarrow{x_J})^2$$

$$= \sum_{j:\overrightarrow{x_J}\in N_i^O} (\overrightarrow{x_\iota} - \overrightarrow{x_J})^T \mathrm{W}W^T\ (\overrightarrow{x_\iota}, -\overrightarrow{x_J})$$

$$= \sum_{j:\overrightarrow{x_J}\in N_i^O} \mathrm{trace}(W^T\ (\overrightarrow{x_\iota}, -\overrightarrow{x_J})\ (\overrightarrow{x_\iota} - \overrightarrow{x_J})^T W)$$

$$= trace\{W^T \cdot \sum_{j:\overrightarrow{x_J}\in N_i^O}\ (\overrightarrow{x_\iota}, -\overrightarrow{x_J})\ (\overrightarrow{x_\iota} - \overrightarrow{x_J})^T \cdot W\} \qquad (10)$$

$$S_i = trace\{W^T \cdot \sum_{j:\overrightarrow{x_k}\in N_i^O}\ (\overrightarrow{x_\iota}, -\overrightarrow{x_J})\ (\overrightarrow{x_\iota} - \overrightarrow{x_J})^T \cdot W\} \qquad (11)$$

$$E_i = trace\{W^T \cdot \sum_{j:\overrightarrow{x_l}\in N_i^O}\ (\overrightarrow{x_\iota}, -\overrightarrow{x_J})\ (\overrightarrow{x_\iota} - \overrightarrow{x_J})^T \cdot W\} \qquad (12)$$

Put Eq.(10), (11), (12) into Eq. (8)

Then J can be expended as

$$J = trace(W^T \cdot (\boldsymbol{\sigma}\Sigma_o + (1-2\boldsymbol{\sigma})\Sigma_s + (\boldsymbol{\sigma}-1)\Sigma_E) \cdot W) \qquad (13)$$

and,

$$\Sigma_O = \sum_{i=1}^{n} \sum_{j:\overrightarrow{x_j} \in N_i^O} (\overrightarrow{x_i}, -\overrightarrow{x_j}) \quad (\overrightarrow{x_i} - \overrightarrow{x_j})^T \qquad (14)$$

$$\Sigma_S = \sum_{i=1}^{n} \sum_{k:\overrightarrow{x_k} \in N_i^O} (\overrightarrow{x_i}, -\overrightarrow{x_k}) \quad (\overrightarrow{x_i} - \overrightarrow{x_k})^T \qquad (15)$$

$$\Sigma_E = \sum_{i=1}^{n} \sum_{l:\overrightarrow{x_l} \in N_i^O} (\overrightarrow{x_i}, -\overrightarrow{x_l}) \quad (\overrightarrow{x_i} - \overrightarrow{x_l})^T \qquad (16)$$

Hence the distance metric learning problem can be formulated as

$$min_{W:W^TW=I} trace(W^T \cdot (\boldsymbol{\sigma}\Sigma_O + (1-2\boldsymbol{\sigma})\Sigma_S + (\boldsymbol{\sigma}-1)\Sigma_E) \cdot W) \qquad (17)$$

As[12] mentioned, for reducing the information redundancy among different dimensions of W, we define orthogonality constraint W^TW = I , as well as control the scale of W to avoid some arbitrary scaling.
Now we need define three symmetric square matrices

DEFINITION 2.4. (Homogeneous Adjacency Matrix) The homogeneous adjacency matrix $\boldsymbol{H^O}$ is an n×n symmetric matrix with its (i, j)th entry

$$h_{ij}^O = \begin{cases} 1, if\ x_j \in N_i^O\ or\ x_i \in N_j^O \\ 0,\ otherwise \end{cases} \qquad (18)$$

DEFINITION 2.5. (Similar Adjacency Matrix) The similar adjacency matrix $\boldsymbol{H^S}$ is an n×n symmetric matrix with its (i, j)th entry

$$h_{ij}^S = \begin{cases} 1, if\ x_j \in N_i^S\ or\ x_i \in N_j^S \\ 0,\ otherwise \end{cases} \qquad (19)$$

DEFINITION 2.6. (Heterogeneous Adjacency Matrix) The heterogeneous adjacency matrix $\boldsymbol{H^E}$ is an n×n symmetric matrix with its (i, j)th entry

$$h_{ij}^e = \begin{cases} 1, if\ x_j \in N_i^e\ or\ x_i \in N_j^e \\ 0,\ otherwise \end{cases} \qquad (20)$$

Following[12], we define $g_{ii}^O = \sum_j h_{ij}^O$ and $\boldsymbol{G^O}$=diag(g_{11}^O, g_{22}^O, g_{33}^O, ⋯, g_{nn}^O)². Likewise, and define $g_{ii}^S = \sum_j h_{ij}^S$, $\boldsymbol{G^S}$=diag(g_{11}^S, g_{22}^S, g_{33}^S, ⋯, g_{nn}^S)², and $g_{ii}^E = \sum_j h_{ij}^E$, $\boldsymbol{G^E}$=diag(g_{11}^E, g_{22}^E, g_{33}^E, ⋯, g_{nn}^E)².

Then we define *Laplacian* Matrix $\boldsymbol{L^O}$, $\boldsymbol{L^S}$, $\boldsymbol{L^E}$:

$$L^O = G^O - H^O \qquad (21)$$

$$L^S = G^S - H^S \quad (22)$$

$$L^E = G^E - H^E \quad (23)$$

As the above definition 2.4, 2.5, 2.6, we can rewrite Eq.(2),(3),(4) as

$$O_i = \sum_{i=1}^{n} O_i = 2trace(W^T X L^O X^T W) \quad (24)$$

$$S_i = \sum_{i=1}^{n} S_i = 2trace(W^T X L^S X^T W) \quad (25)$$

$$E_i = \sum_{i=1}^{n} E_i = 2trace(W^T X L^E X^T W) \quad (26)$$

Then put Eq.(24), (25), (26) into Eq.(10),and the optimization problem becomes

$$\boldsymbol{min}_{\boldsymbol{W^T W = I}} \boldsymbol{Trace(W^T \cdot X \cdot (\sigma L^O + (1-2\sigma) L^S + (\sigma - 1) L^E) \cdot X^T \cdot W)} \quad (27)$$

With the following *Ky Fan* theorem[13], we know the above solution would be $\mathbb{W}=[\boldsymbol{w_1, w_2, \ldots, w_n}]$, and $[\boldsymbol{w_1, w_2, \ldots, w_n}]$ are the eigenvectors of matrix $\boldsymbol{X \cdot (\sigma L^O + (1-2\sigma) L^S + (\sigma - 1) L^E) \cdot X^T}$,whose corresponding eigenvalues are negative.

In the end, Put **W** into Eq.(8), then put Eq.(8) in to Eq. (1), we get the distance metric.

The summary of MSML algorithm as shown on Table.3.1:

Input	Process	Output
1. Data matrix **X** **2.** Label group **Y** **3**. Gap's weighting parameter: $\boldsymbol{\sigma}$	**1.** Construct Homogeneous , Similar and Heterogeneous Neighbour: $\boldsymbol{N_i^o}$, $\boldsymbol{N_i^S}$, $\boldsymbol{N_i^E}$ using **Eq.** (2) and **Y**. **2.** Construct $\boldsymbol{H^O}$, $\boldsymbol{H^S}$, $\boldsymbol{H^E}$ using **Eq.** (18), (19), (20). **3.** Construct Homogeneous , Similar and Heterogeneous *Laplacian*: $\boldsymbol{L^O, L^S, L^E}$ using **Eq**. (21), (22), (23). **4**. Figure out the matrix formula $\boldsymbol{X \cdot (\sigma L^O + (1-2\sigma) L^S + (\sigma - 1) L^E) \cdot X^T}$, and the each value from large to	**1.**Distance metric: $\boldsymbol{D_M(\overrightarrow{x_i}, \overrightarrow{x_j})}$

	small as **d-**dimensional matrix of **W.** **5**. Figure out $\boldsymbol{D_M(\overrightarrow{x_i}, \overrightarrow{x_j})}$ Using **Eq.** (9), (1).	

Table.3.1 MSML algorithm

The gap's weighting parameter: $\boldsymbol{\sigma}$ $[0.1 - 0.9]$ need to be trained and test in MSML algorithm.

3.2 Bidirectional Heuristic Search

In order to meet condition as Session 2.2 Step 5, my algorithm has following features.

- Use modified bidirectional A* heuristic search
- The heuristic function is dynamic weighting
- Front-to-Front heuristic, new expansive node to be new goal node
- The expansion node is feature readable
- The new start node is dynamic changing the rule of open list collection

In contrast, the grid map problem could solve by traditional A*, the nodes are distributed in gird, and it collects the surrounding eight nodes to the open list. In our case, each node is unique, and irregular and spread out, as shown on **Session 2.2, Step 1.** The only unified standard is the Label set **Y** which we defined for the set **X.** As shown on **CASE 1.1**, even though a person associated two irrelevant words, at least we can start from the positive and negative emotion to carry out analysis. In this case, there is only one kind of emotion label which built for the similar degree of each word, but in some complex question, there are multi emotions on a single word, to measure each word's metric distance with multi-label in a database, the MSML algorithm is a solution.

After we got the distance, how to collect the near nodes? Because there are too many irrelevant nodes, so the question come back to the original puzzle, how a person achieves the self-association? whether or not there is totally no relationship between these two words. Following [2], human cognitive and associational methods are based on the feature of the object. Hence, our goal is to collect some nodes with the same feature from opposite node.
Because the Heuristic function has to use my MSML algorithm, so I called this Bidirectional Front-to-Front Iterative Dynamic Weighting Heuristic Search (**BFIDWHS-MSML**). BFIDWHS-MSML is for generating rational expansions.

Definitions and terminology

Forward search:	**Backward search:**
$f_F(n)=\min g_F(n', n) + w_F(F_F)*h_F(n, m')$,	$f_B(m)=\min g_B(m, m') + w_B(F_B)*h_B(m, n')$

The first n' and m' are Start and Goal node	The first n' and m' are Start and Goal node
$\mathbf{g_F}$: MSML heuristic function, from forward expansive node **n** to previous node **n'**,	$\mathbf{g_B}$: MSML heuristic function, from backward expansive node **m** to previous node **m'**,
$\mathbf{h_F}$: MSML heuristic function, from expansive node **n** to previous backward node **m'**,	$\mathbf{h_B}$: MSML heuristic function, from expansive node **m** to previous forward node **n'**,
$\mathbf{w_F}$: the weight of heuristic function.	$\mathbf{w_B}$: the number of common features.
$\mathbf{F_F}$: the number of common features.	$\mathbf{F_B}$: the number of common features.
Start is start node, as	**Goal** is goal node, as
n is forward expansive node	**m** is backward expansive node
n' is previous forward node	**m'** is previous backward node
$\mathbf{Open_F}$: a list for the nodes to scan.	$\mathbf{Open_B}$: a list for the nodes to scan.
$\mathbf{Closed_F}$: a list for expansive nodes.	$\mathbf{Closed_B}$: a list for expansive nodes.

Core Algorithm:

*The first n' and m' are Start and Goal node.

Backward search: $f_B(m)=\min g_B(m', m) + w_B(F_B)*h_B(m, n')$

$\mathbf{Open_B}$ = Open list, $\mathbf{Closed_B}$ =Close list;

Forward search: $f_F(n)= \min g_F(n', n) + w_F(F_F)*h_F(n, m')$,

$\mathbf{Open_F}$ = Open list, $\mathbf{Closed_F}$ =Close list.

This algorithm uses four lists, starts from Backward search, stop by the first collision between two Close lists. The data input is the data set X, x_i is arbitrary subset to set X. **xi(s)** means multiple arbitrary subsets. and label set Y, y_i is arbitrary subset to set Y.

the open list collection utilizes KNN-MSML method, each time the previous node **n'** or **m'** use KNN-MSML to search the k number of closest neighbors, and compare with the opposite previous node **m'** or **n'**, find out minimum $\mathbf{F_F}$ or $\mathbf{F_B}$ number of common feature **min f_i(n) n>1**. Measure all the distance between $\mathbf{x_n}$' or $\mathbf{x_m}$' to the set **X**, put all of the subset **x_i(s) with min f_i** common feature in to open list, $\mathbf{w_F}$ **or** $\mathbf{w_B}$ scale up following $\mathbf{F_F}$ or $\mathbf{F_B}$, Min $g_{F \text{ or } B} + h_{F \text{ or } B}$ make sure the direction forward to goal, put the $\mathbf{f_B(m)}$ or $\mathbf{f_F(n)}$ to the close list, and remove all of the subset **xi(s)** out of set **X**. Loop until the first collision.

3.2.1 Algorithm explanation:

As shown on **Fig 3.2**, [3] is Start node, [9] is Goal node, [14] is **n** or **n'** node, [20] is m or m' node, [17] and [18] are the middle nodes. [4] is Forward search, [19] is backward search, [12] is the data set X and label set Y.

Step 1: Backward search

Because the **Start** node is given condition, **Goal** node [9] is chosen by participant, the first step should find one common feature with the **Start** [3]. Set Goal node as $\mathbf{x_G}$, and $\mathbf{x_G}$ need to one-to-one correspondence all the subsets from set **X**. But this is only time a subset pairs the whole data set X. k number would increase until there is a neighbor $\mathbf{x_i}$ has at least one common feature fi with Start node's feature. $\mathbf{F_B}$ is the number of **fi**. [10] is the distance from this neighbor $\mathbf{x_i}$ to **Goal.** but we need to keep increase the k number, because **xi** might located further or even opposite direction with **Backward** [19]. Hence, until $\mathbf{F_B}$ > the first $\mathbf{F_B}$**, KNN** stop searching.

[11] is the metric distance which including k+n number of the goal node's neighbors. and then put all the nodes with $\mathbf{F_B}$ feature into $\mathbf{Open_B}$ List. Using $f_B(m)=\min g_B(Goal, m) + w_B(F_B)*h_B(m, Start)$ to traverse $\mathbf{Open_B}$, at last put $\mathbf{f_B(m) = m}$ node [20] into $\mathbf{Closed_B}$. and remove all the $\mathbf{Open_B}$ nodes from the set **X,** because we don't need to pair them anymore. if $\mathbf{F_B=1}$, $\mathbf{w_B}$ = 1, when $\mathbf{F_B=F_B+1}$, $\mathbf{w_B}$ =1 + 0.2. Because the closer, the more synonyms between **n'** and **m',** so it needs acceleration to increase efficiency.

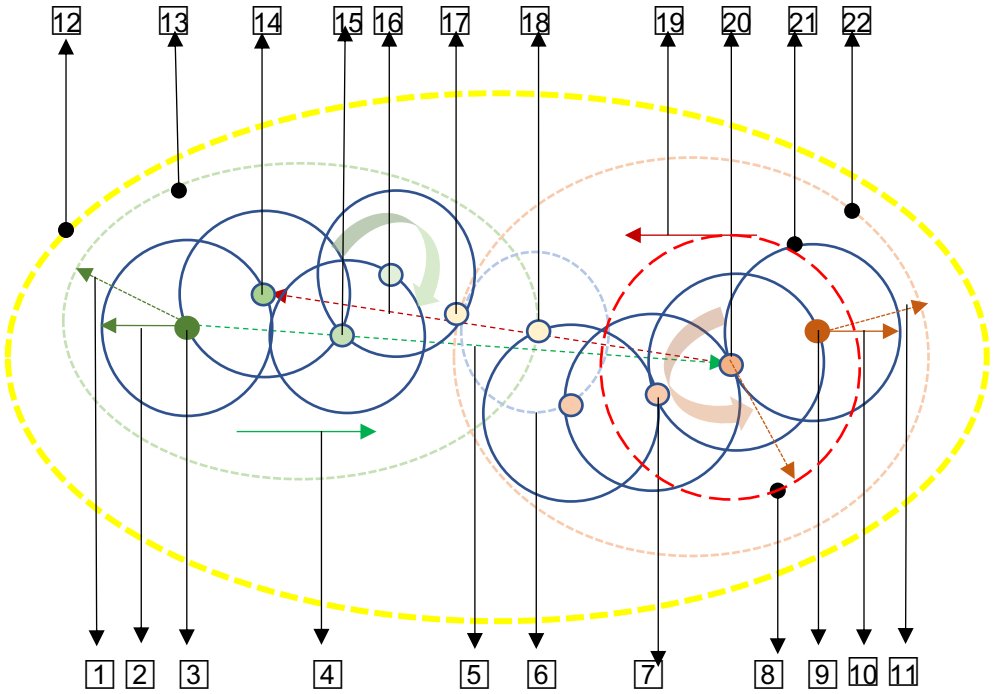

Fig.3.2 BFIDWHS-MSML

In **CASE 1.1,** this **Step 1** would find out a word "future" which contains fi = "bright", is the same with the feature in "cloud".

Step 2: Forward search

This step is basically the same with **Step 1,** but the **Start** node [3] finds one more common feature with the **m'** [20]. Using $f_F(n)=\min g_F(Start, n) + w_F(F_F)*h_F(n, m')$ to traverse **Open$_F$**, at last put **$f_F(n) = n$** node [14] into **Closed$_F$**.

In **CASE 1.1,** this **Step 2** would find out a word "sunset" which contains fi = "time", is the same with the feature in "future".

Step 3: Iterative Backward search

Turn back to **Backward search** the **m'** [20] finds one more common feature with the **n'** [14], $f_B(m)=\min g_B(m', m) + w_B(F_B)*h_B(m, n')$ to traverse **Open$_B$**, at last put **$f_B(m) = m$** node [7] into **Closed$_B$**.

In **CASE 1.1,** this step would find out a word "tomorrow" which contains fi = "sunshine", is the same with the feature in "sunset".

Step 4: Iterative Forward search

Go to **Forward search**, The **n'** node [14] finds at least one more common feature with the **m'** [7]. Using $f_F(n)=\min g_F(n, n) + w_F(F_F)*h_F(n, m')$ to traverse **Open$_F$**, at last put **$f_F(n) = n$** node [15] into **Closed$_F$**.

In **CASE 1.1,** this step would find out a word "drown" which contains fi = "alternate", is the same with the feature in "tomorrow".

Until the Middle nodes [17] or [18] in both **Closed$_F$** and **Closed$_B$**, terminate the search.

3.2.2 Algorithm 3.2: BFIDWHS-MSML

1. **Input**: Data set **X_i**, Label set **Y_i,** Gap's weighting parameter: $\boldsymbol{\sigma}$, **Start**, **Goal**;
2. **Output**: **Closed$_F$**(n),. **Closed$_B$**(m).
3. (**x_S**, **x_G**) ∈**X_i ; g_F**(Start) := **x_S**; **g_B**(Goal):= **x_G**;
   f_s ∈ **x_S**, **f_G** ∈ **x_G; F_F** := **F_B** := 1; **$W_F(F_F)$** := **$W_{B(F_B)}$** := 1;
4. Calculate Metric Distance $D_M(\overrightarrow{x_i}, \overrightarrow{x_j})$ between **x_G** with ∀ **Xi;**
5. Generate distance set **h (x_G, ∀Xi)**
6. while (**f_i** ∈ ∃(**$x_m(n)$**), n>1,n=n+1, **f_s** ∈ **x_S**, **$f_i = f_s$**, **f_i +1 > fi**)

do $Open_B \leftarrow$ **$x_m(n)$ (n>1),**

7. while ($Open_B = \varnothing$)
   if $f_B(m)$= min $g_B(m, Goal) + w_B(F_B)h_B(m,Start)$
   $Closed_B \leftarrow f_B(m)$;
   Remove **∀ $x_m(n)$** from **Xi;**
   If ($Closed_F \cap Closed_B$) //Terminal condition
   Return Announce (GoalFound)
8. Calculate Metric Distance $D_M(\overrightarrow{x_i}, \overrightarrow{x_j})$ between **x_S** with **∃Xi ⊄ $Open_{B'}$;**
9. Genrate distance set **h (x_S, ∃Xi)**
10. while (**f_i** ∈ ∃(**$x_n(n)$**), n>1,n=n+1, **$f_{m'}$** ∈ **$x_{m'}$**, **$f_i = f_{m'}$**, **f_i +1 > fi**)
    do $Open_F \leftarrow$ **$x_n(n)$ (n>1), $F_F=F_F$ +1, W_F =1+ 0.2;**
    if $f_F(n)=min\ g_F(n, Goal) + w_F(F_F)h_F(n,m)$
    $Closed_F \leftarrow f_F(n)$;
    Remove **∀ $x_n(n)$** from **Xi;**
    If ($Closed_F \cap Closed_B$) // Terminal condition
    Return Announce (GoalFound)
11. Calculate Metric Distance $D_M(\overrightarrow{x_i}, \overrightarrow{x_j})$ between **$x_{m'}$** with **∃Xi⊄ $Open_{F'}$;**
12. Generate distance set **h (x_m, ∃Xi)**
    Clear **$Open_B$;**
13. while (**f_i (n)**∈ ∃(**$x_m(n)$**), n>1,n=n+1, **$f_{n'}$** ∈ **$x_{n'}$**, **$f_i = f_{n'}$**, **f_i +1 > fi**)
    do $Open_B \leftarrow$ **$x_m(n)$ (n>1), $F_B=F_B$ +1, W_B =1+ 0.2;**
14. while ($Open_B = \varnothing$)
    if $f_B(m)=min\ g_B(m, m') + w_B(F_B)h_B(m,n')$
    $Closed_B \leftarrow f_B(m)$;
    Remove **∀ $x_m(n)$** from **Xi;**
    If ($Closed_F \cap Closed_B$) //Terminal condition
    Return Announce (GoalFound)
15. Calculate Metric Distance $D_M(\overrightarrow{x_i}, \overrightarrow{x_j})$ between **x_n** with **∃Xi⊄ $Open_{B'}$;**
16. Generate distance set **h (x_n, ∃Xi)**
    Clear **$Open_F$;**
17. while (**f_{i++}** ∈ ∃(**$x_n(n)$**), n>1,n=n+1, i=i+1, **$f_{m'}$** ∈ **$x_{m'}$**, **$f_i = f_{m'}$**, **f_i +1 > fi**)
    do $Open_F \leftarrow$ **$x_n(n)$ (n>1), $F_F=F_F$ +1, W_F =1+ 0.2;**
18. while ($Open_F = \varnothing$)
    if $f_F(n)=min\ g_F(n,n') + w_S(F_F) * h_F(n,m')$
    $Closed_F \leftarrow f_F(n)$;
    Remove **∀ $x_n(n)$** from **Xi;**
    If ($Closed_F \cap Closed_B$) //Terminal condition
    Return Announce (GoalFound).

4. Conclusion

This paper focus on solving two challenges, the first is to redefine every different kinds of noun, put corresponding adjective and quantized its feature. Try to quantize emotion to be the label of noun, propose a generalized *Mahalanobis* metric distance learning method: MSML method to find out the similarity between different sample with multi-labels. The other is that propose a **BFIDWHS-MSML** algorithm which is using MSML to be the heuristic function, for find out the common feature from opposite node and expansive the new node, and then generate an associational path. In the end, through increasing each common feature weight to re-generate associational path, the fewest nodes expansion path representing the key associational word between the start and goal word.

5. Limitation and discussion

The database construction is a long-term work, and it still need to cooperate with the expert from psychological area and linguistic area. But the most important part is to continue and accumulate some experience, there is only “noun” to be the subset **xi**, “verb” is also the features in **xi,** in order to unify and quantize “verb”, we use “location” + “time” + “relevant feature transformation” to describe a verb on a noun, at the beginning, the test scenario focus on children education, it will be easy to build the original database.

The MSML algorithm has optimal solution in math, and the paper from Ni Jiazhi [14] propose a similar idea, even though they did not show any details of their distance measure method, at least, it’s accepted that the gap pushing concept which is making the samples with similar labels be closer.

The **BFIDWHS** algorithm is a prototype to build the expansive nodes path, but not for searching the optimal path. There are lots of way that we can add to search the feature on **BFIDWHS**, when we build enough paths or generate enough rational expansive nodes, then we can optimize **BFIDWHS** to find out the optimal path.

References

[1]. Laughlin Jr, Charles D., John McManus, and Eugene G. Brain, symbol & experience: Toward a neurophenomenology of human consciousness. Columbia University Press, 1992.

[2]. Jeff Hawkins, Sandra Blakeslee, On Intelligence: How a New Understanding of the Brain will Lead to the Creation of Truly Intelligent Machines. 2004:Chapter 4,81.

[3]. I. Pohl, Bi-directional and heuristic search in path problems. PhD thesis, Dept. of Computer Science, Stanford University., 1969.

[4]. Korf, R. "Depth-first iterative deepening : An optimal admissible tree search," Artificial Intelligence, 27(1), 1985.

[5]. Arefin, K. S., & Saha, A. K. (2010). A new approach of iterative deepening bi-directional heuristic front-to-front algorithm (IDBHFFA). International Journal of Electrical and Computer Sciences (IJECS-IJENS), 10(2).

[6]. Holte, Robert C., et al. "Bidirectional search that is guaranteed to meet in the middle." Thirtieth AAAI Conference on Artificial Intelligence. 2016.

[7]. Sint, L., and de Champeaux, D. 1977. An improved bidi- rectional heuristic search algorithm. Journal of the ACM (JACM) 24(2):177–191.

[8]. Mayer, Leopold E., and Kurt D. Krebsbach. "Front-to-Front Bidirectional Best-First Search Reconsidered." (2019).

[9]. M. Lippi, M. Ernandes, and A. Felner, “Efficient single frontier bidirectional search.,” in SOCS, 2012.

[10]. Weinberger KQ, Saul LK. Distance metric learning for large margin nearest neighbor classification. Journal of Machine Learning Research, 2009, 10: 207–244.

[11]. Sun J, Wang F, et al. Supervised patient similarity measure of heterogeneous patient records. ACM SIGKDD Explorations Newsletter, 2012, 14(1): 16–24.

[12]. F. Wang, J. Sun, T. Li, and N. Anerousis. Two heads better than one: Metric+active learning and its applications for it service classification. In *IEEE International Conference on Data Mining*, pages 1022–1027, 2009.

[13]. Zha, Hongyuan, et al. "Spectral relaxation for k-means clustering." Advances in neural information processing systems. 2002.

[14]. Ni, Jiazhi, et al. "Fine-grained patient similarity measuring using deep metric learning." Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. ACM, 2017.